

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method for visualizing any architecture during conceptual, development or and deployment phases of a system, said method comprising the steps of:

receiving information regarding ~~a modeled architecture~~ an event;

determining ~~the a~~ look of ~~the an~~ application;

determining ~~the a~~ feel of the application;

~~receiving requests from the event service implementation and delegating these requests to the event service implementation;~~

receiving events ~~from the modeled architecture and~~ at an event service delivery agent;

forwarding ~~these the~~ events to a model component of the architecture in the form of callbacks ~~on the modeled architecture~~;

reading configuration information during initialization at application startup, wherein the configuration information is adaptable to a plurality of different middlewares; and

presenting the associated behavior of the system.

2. (Currently Amended) A The method of claim 1, wherein, ~~in~~ said step of presenting[[.]] comprises the step of presenting said architecture ~~is presented~~ in simulation mode.

3. (Currently Amended) A The method of claim 1, wherein, ~~in~~ said step of ~~presented~~, presenting comprises the step of presenting said architecture ~~is presented~~ in auto-cycle a continuous, free-running mode.

4. (Currently Amended) A The method of claim 1, wherein, ~~in~~ said step of ~~presented~~, presenting comprises the step of presenting a deployed implementation of the architecture ~~is presented~~.

5. (Currently Amended) A The method of claim 1, wherein said configuration information further comprises:

abstract information in the form of tiers, components, communication paths and events[[.]];

presentation information in the form of how many display views are required to present the architecture, and how to respond visually when events are received;

controller information that may specify details that determine how the particular controller implementation behaves; and

integration information that may be used by the particular implementation of an event service delivery agent.

6. (Currently Amended) An architecture visualization system for visualizing any architecture during conceptual, development or deployment phases of a system, said architecture visualization system ~~further~~ comprising:

- a model component for receiving information regarding events ~~the modeled architecture~~;
- a view component for determining ~~the~~ a look of an application;
- a controller for effectively determining ~~the~~ a feel of the application;
- an event service delivery agent for receiving and delegating requests; and
- a configuration specified in XML, wherein the configuration is adaptable to a plurality of different middlewares.

7. (New) A system for visualizing an application architecture, the system comprising:

- an event service interface for receiving middleware or protocol dependent events, the event service interface for translating the middleware or protocol dependent events into middleware or protocol independent events;

- an event service delivery agent for receiving the middleware or protocol independent events and managing delivery of events to an application;

- a model component for receiving middleware or protocol independent events from the event service delivery agent;

- a view component receiving middleware or protocol independent events from the model component, the view component for visualizing behavior of an application during at least one of conception, development, and deployment of the application; and

- a controller for managing the behavior of the application, an implementation of the controller being adaptable based on the middleware communicating with the application.

8. (New) The system of claim 7, further comprising an XML configuration for creating implementations of the event service delivery agent and the controller, wherein the implementations are based on the middleware communicating with the application.

9. (New) The system of claim 8, wherein the controller manages behavior of the application based on at least one of information received from the view component, the model component, and XML configuration information related to the controller.

10. (New) The system of claim 8, wherein the XML configuration comprises:
abstract information in the form of tiers, components, communication paths and events;
presentation information in the form of how many display views are required to present the architecture, and how to respond visually when events are received;
controller information that may specify details that determine how a particular controller implementation behaves; and
integration information that may be used by a particular implementation of an event service delivery agent.

11. (New) The system of claim 7, wherein the view component creates different views of the application dependent on whether the system is in a conception mode, a development mode, or a deployment mode.

12. (New) The system of claim 7, wherein the view component deploys a slowed down event visualization in a demonstration mode.

13. (New) The system of claim 7, wherein the view component creates a visualization based on configuration information.

14. (New) The system of claim 7, wherein the event service delivery agent forwards events to the model component as a callback instruction.